

## \$ # k **Hotkeys version 1.3**

Give your poor mouse a break with Hotkeys. Hotkeys is similar to the Microsoft Intellitype software, in that it gives you fast and easy access to a lot of common Windows 95 elements through keyboard shortcuts.

Hotkeys does this and a lot more. Use it to define hotkeys to launch applications such as Control Panel applets or frequently used programs; send keystrokes to other applications; exit Windows, log-off Windows, or reboot your computer; switch to one of Hotkeys' built-in virtual clipboards; or quickly access some of Hotkeys' internal functions such as its editor or hotkey list. Hotkeys works with nearly every key on your keyboard, including that seldom-used Windows key. You can even instruct Hotkeys to execute multiple Hotkeys commands sequentially, making it the ideal tool for scripting tasks you have to perform often.

[How To Use Hotkeys](#)

[What's new in version 1.3?](#)

[What are virtual clipboards?](#)

[Hotkeys program and source-code](#)

[Postcardware](#)

Join the Hotkeys mailing list at <http://hotkeys.listbot.com> if you'd like to be informed about future versions of Hotkeys.

## \$<sup>#</sup><sup>k</sup>How to use Hotkeys

Simply run the supplied installation program (SETUP.EXE). This will copy the Hotkeys program, the sample hotkey definition file ([Hotkeys.hkd](#)), the Hotkeys helpfile (this file) and two utility files (HKRESTRT.EXE and HKTODOS.PIF) to the directory C:\Program Files\Hotkeys. It also starts the Hotkeys application (you can see it appear as a key with an H on it in the system tray). The installation program also adds an entry in the registry that causes Hotkeys to be started every time you start Windows.

If you press a hotkey defined in the hotkey definition file, Hotkeys will perform the associated action (e.g. starting an application, sending keystrokes etc). You can see that Hotkeys works by looking at the tray-icon: an animation of a key being pressed is played every time you press a registered hotkey.

To edit the hotkey definitions, right click on the Hotkeys tray-icon and select **Edit hotkeys** from the popup-menu or press **Win+H**. This will bring up the [Hotkey editor](#), where you can add, edit or delete hotkey definitions.

To view the Hotkey definitions, select **View hotkey list** from the popup-menu or press **Win+V**. The [Hotkey list](#) window displays all active hotkeys. To perform the action (like starting an application or sending keystrokes) associated with the hotkey double-click on an entry in the Hotkey list or select an entry with the arrow keys and press Enter.

The Hotkeys popup-menu also contains a list of the defined programs and their associated hotkeys under the **Hotkeys** menu item.

## \$ \$# #k The Hotkeys program and source-code

Hotkeys was written entirely in Borland's Delphi (version 3.02) and comes with complete source-code. The source code demonstrates a lot of Win95/NT features like tray notification, animated icons, registering system global hotkeys etc. Most of this functionality is encapsulated in the following components:

AniIcons	Storage for animated icons
AniPlay	Animated icon player (used in Hotkeys about box)
AniTray	Animated tray-icon
SysHot	Component for registering and trapping hotkeys

You can find the source code in the following directories:

Source\	Source code for the Hotkeys program
Source\Components\	Source code for the component package
Source\Helpfile\	Source code for this helpfile
Source\Animations\	Icons and animation files used in the Hotkeys program
Source\Setup	Source code for the setup program
Source\HKRestart	Source code for HKRESTRT program

If you want to take a look at the Hotkeys source-code, make sure that you install the components first. Installing the components is easy:

If you use Delphi 3 or 4, open the package file Shear.dpk from the **Source\Components** directory click the **Install** button. For Delphi 2 select **Install** from the **Components** menu and install the files **AniReg.pas**, **BrowseEdit.pas** and **SysHot.pas**.

## \$ \$ #<sup>K</sup>PostcardWare

The Hotkeys program and the accompanying source-files are PostcardWare (which essentially stands for: if you like this software, send me a Postcard from Ware you live). This means that you're completely free to use or modify it to suit your needs, with just a few restrictions:

- The original copyright remains with the author of the software.
- It is not allowed to sell this package as a whole pretending you wrote it or otherwise distribute it under your own name.
- You are encouraged to distribute this package to anybody you like, but you have to give them the entire archive unmodified.
- You're not allowed to ask money for the distribution of this package.
- If you use (portions of) the source-code in your own programs, you don't have to mention my name in the credits or about-box. You do have a moral obligation to send me a postcard of your hometown however. Please tell me where you found Hotkeys and what you think of the program. Bug-reports and suggestions for improvements are also welcomed.
- If you don't use the source-code, but you continue to use the Hotkeys program, I feel that I'm also entitled to a postcard of your hometown.
- Unless there's a critical error with the software, I usually only respond to e-mails about the software if your postcard is in the mail.

Send those postcards to:

SheAr software  
Arjen Broeze  
Gronausevoetpad 104  
7511 BN Enschede  
the Netherlands  
e-mail: Arjen@Earthling.net

## \$<sup>#</sup> #<sup>k</sup> What are virtual clipboards?

As of version 1.3, Hotkeys has virtual clipboards. A virtual clipboard is nothing more than a storageplace for clipboard data. You can put the data from a virtual clipboard onto the Windows clipboard by pressing a hotkey (the hotkey to use for switching to a particular virtual clipboard can be defined in the [Hotkey Editor](#). By default Hotkeys has 5 virtual clipboards, but you can specify any number (up to 99) in the Hotkey Editor.

### How do virtual clipboards work?

When Hotkeys starts, virtual clipboard #1 is active. You can see this by looking at the Hotkeys icon in the system tray. In the lower right corner you see the number 1. If there is no data on the clipboard, the number of the active clipboard is painted in a dark blue circle, {bmc clip1a.bmp} otherwise it is painted in a bright blue circle {bmc clip1b.bmp}.

When you switch to virtual clipboard #2 the data that is currently on the Windows clipboard is copied into virtual clipboard #1. Unlike other virtual clipboards who can only copy text and graphics, Hotkeys copies all the data on the Windows clipboard to the virtual clipboard no matter which format it is in. After the clipboard data has been copied, the data on virtual clipboard #2 is put on the Windows clipboard. If you're using the default hotkey definition file you can switch to virtual clipboard #2 by pressing Win+2. Since every virtual clipboard is initially empty, the Windows clipboard will be emptied when you switch to a virtual clipboard for the first time. Now you can copy new data to the clipboard and use it the Windows way (Ctrl-C, X, and V). When you switch back to virtual clipboard #1, the data on the Windows clipboard is copied into virtual clipboard #2 and the data in virtual clipboard #1 is put back on the clipboard ready to be used. That's all there is to it.

### Why virtual clipboards?

Windows only has one clipboard. When you copy something to the clipboard, the original data on the clipboard is lost. This is annoying if you need to copy something to the clipboard, but you also want to retain what's on the clipboard already. Virtual clipboards solve this shortcoming. Simply switch to another clipboard, copy and paste the other data and switch back to the original previous virtual clipboard.

\$ What are virtual clipboards?

# IDD\_VIRTCLIP

<sup>k</sup> Virtual clipboards; Adding hotkeys; Hotkeys editor

## \$<sup>#</sup> <sup>k</sup> The Hotkey Editor

The Hotkey editor lets you add, edit or delete hotkey definitions. When you are editing hotkeys and there are unsaved changes (the Apply button is enabled), all defined hotkeys are disabled. As soon as you press the Apply button or close the editor window the hotkeys are enabled again.

You can get more information on the hotkey editor by clicking on parts of the picture below:

{bmc editor.shg}

\$ The Hotkey editor

# IDD\_EDITOR

<sup>k</sup> Adding hotkeys; Editing hotkeys; Hotkeys editor; Sending keys to applications



# The hotkey definition list shows the definitions of all programs and their associated hotkeys. To edit an existing definition, simply click on it. Every change you make to a new or existing hotkey definition is updated in the list immediately. To save the changes you've made and activate new hotkeys, press the **Apply** or **Close** button.



#Type the description for the program you want to run in this field. The description you type here is used in the Hotkeys menu-item in the tray-icon popup-menu.

# This field holds the ID for this action. It is not necessary to type an ID here. Hotkeys generates a unique ID from the description you type in the Description-field. ID's are only needed for the Execute Multiple Commands action.

# Here you have to choose the action for this particular hotkey definition. The following actions are possible:

Execute Program	Starts the application specified in the 'Command line' edit box.
Execute Multiple Commands	Allows you to execute multiple Hotkeys commands sequentially.
Send keystrokes	Sends the keys you typed in the 'Keys to Send' edit-box to the active application. Click <a href="#">here</a> for a list of special keys.
Exit Windows	Depending on the chosen parameter in the 'Parameter' combo-box, this command will either shutdown, restart or reboot Windows, exit to DOS, log the current user off or suspend the computer
Display About Box	Displays the About window. This is the same as selecting 'About' from Hotkeys tray-icon popup-menu.
Edit Hotkeys	Displays the Hotkey editor. This is the same as selecting 'Edit Hotkeys' from the tray-icon popup-menu.
View Hotkeys	Displays the Hotkey list. This is the same as selecting 'View Hotkey list' from the tray-icon popup-menu.
Hotkeys Help	Displays the contents topic of this Helpfile.
Minimize All Windows	Minimizes all open windows.
Undo Minimize All	Restores previously minimized windows.
Switch to Virtual Clipboard	Copies the data from one of Hotkeys virtual clipboards to the Windows Clipboard.

# IDD\_EDIT\_ID

# IDD\_EDIT\_ACTION

# Depending on the chosen action above you can either specify the name of the program to run followed by possible parameters, type the keys that have to be sent to another application, type the IDs of the actions you want to be executed separated by semi-colons (;), select what action to perform in the 'Parameters' combo box or select the number of the virtual clipboard to switch to.



# If the action you've chosen is 'Execute Program', clicking on this button allows you to browse your disks to locate a program you want to associate with the defined hotkey. If the action you've chosen is 'Execute Multiple Commands', clicking this button takes you to the Select Commands dialog where you can select the commands that need to be executed.

# This combobox allows you to specify how a program has to be started. A program is started normal by default, but you can also select 'Minimized' to start a program minimized to the taskbar or 'Maximized' to start a program maximized to the screen.

# This is where you specify the hotkey for the selected application, set of keystrokes or command. Every time you make a change to a hotkey, the validity of that hotkey is checked because a system-wide hotkey can only be defined once. If some other program already registered the hotkey you're trying to set, Hotkeys will inform you and reset the hotkey to its previous value.

#Uncheck this box if you want to disable a particular hotkey, but would like to retain it in the list of hotkey definitions so you can reactivate it later. If you want to permanently remove a hotkey combination, use the **Delete** button.



# This checkbox determines whether or not Hotkeys is accessible through the Windows 95/NT system tray. Check this box if you want to be able to quickly access the Hotkeys editor. If you uncheck the checkmark, the only way you can access the Hotkeys editor is by pressing the defined hotkey (default Win+H) or by starting the Hotkeys program (Hotkeys.exe).

# Click this button to create a new hotkey definition. When you click the button, a new definition is created and all input fields are cleared. Although the new hotkey definition is added to the list, it will not become active until you press the **Apply** or **Close** button

# Click this button to delete an existing hotkey definition. When you press the Delete button, the hotkey is removed from the list. However, the hotkey will remain active until you press the **Apply** or **Close** button.

#When you click this button, all the changes you have made to the hotkey definitions are saved.

# This button closes the Hotkey Editor. If you have unsaved changes, you will be prompted to save them before closing the Hotkey Edit.

# Here you can specify the number of virtual clipboards Hotkeys can retain. You can set this to any value between 1 and 99. To be able switch to a particular virtual clipboard you must define a hotkey for it.

\$\$\$#<sup>k</sup>{bmc list.bmp}

The Hotkey list window displays all active hotkeys. To perform an action associated with a particular hotkey (like starting an application or sending keystrokes) you can either press the hotkey combination, double-click on an entry in the Hotkey list or select an entry with the arrow keys and press Enter.

# IDD\_EDIT\_NUMCLIPS

\$ The hotkey list window

# IDD\_VIEW

<sup>k</sup> Hotkey list window

## **# The Hotkey list window**

The Hotkey list window displays all active hotkeys. To perform an action associated with a particular hotkey (like starting an application or sending keystrokes) you can either press the hotkey combination, double-click on an entry in the Hotkey list or select an entry with the arrow keys and press Enter.

\$ #<sup>k</sup>The hotkey definition file initially contains the following definitions:

<b>Description</b>	<b>Hotkey</b>
Accessibility Properties	Win+A
Add/Remove Programs	Win+W
Clipboard Viewer	Ctrl+Win+C
Control Panel	Win+C
Date/Time Properties	Win+T
Display Properties	Win+D
Edit Hotkeys	Win+H
Exit to DOS	Ctrl+Win+D
Exit Windows	Ctrl+Win+X
Fonts	Alt+Win+F
International Settings	Win+I
Keyboard Properties	Win+K
Maximize Window	Alt+F2
Minimize Window	Alt+F1
Modem Properties	Ctrl+Win+M
Mouse Properties	Alt+Win+M
Network Settings	Win+N
Password Properties	Ctrl+Win+P
Power Management	Alt+Win+P
Printers	Win+P
Reboot System	Ctrl+Win+R
Restore Window	Alt+F3
Suspend	Ctrl+Win+S
System Properties	Win+S
View hotkey list	Win+V
Virtual Clipboard 1	Win+1
Virtual Clipboard 2	Win+2
Virtual Clipboard 3	Win+3
Virtual Clipboard 4	Win+4
Virtual Clipboard 5	Win+5

\$ The Hotkey definition file

# IDD\_DEFAULT

<sup>k</sup> Hotkey definition file



<sup>\$</sup><sup>#</sup><sup>k</sup>If you like to send special characters as keystrokes, you'll have to enclose them in curly braces. The following table defines the codes you can use to send these special characters:

Key	Code
Backspace	{BACKSPACE} or {BS} or {BKSP}
Clear	{CLEAR}
Caps Lock	{CAPSLOCK} or {CAPS}
Del	{DELETE} or {DEL}
Down Arrow	{DOWN}
End	{END}
Enter	{ENTER} or ~
Esc	{ESCAPE} or {ESC}
Help	{HELP}
Home	{HOME}
Ins	{INSERT} or {INS}
Left Arrow	{LEFT}
Num Lock	{NUMLOCK}
Page Down	{PAGEDOWN} or {PGDN}
Page Up	{PAGEUP} or {PGUP}
Right Arrow	{RIGHT}
Scroll Lock	{SCROLLLOCK} or {SCROLL}
Print screen	{PRINTSCREEN} or {PRTSC}
Tab	{TAB}
Up Arrow	{UP}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}

To specify keys combined with any combination of Shift, Ctrl, and Alt keys, precede the regular key code with one or more of the following codes:

Key	Code
Shift	+
Control	^
Alt	%

To send any of the modifier keys (+, ^, %, ~, { and }), enclose them in curly braces. For example, to make calculator calculate the results of 75% of (23 + 45 + 32) use 75{%}(23{+}45{+}32)~ as the string to send. If everything works, the display should read 75.

It is also possible to specify a delay when sending keystrokes to other applications. To do this insert the text {DELAY=xxxx} anywhere in the string to send, where xxxx is the amount of milliseconds to wait between each character.

<sup>\$</sup> Special character list

<sup>#</sup> IDD\_KEYS

<sup>k</sup> Sending keys to applications; Sending special characters

## \$ \$# #<sup>k</sup>What's new in version 1.3?

The only major new feature in version 1.3 is the use of virtual clipboards. For more information on virtual clipboards and how to use them see the section [What are virtual clipboards?](#)

### What was new in version 1.2?

The only change between version 1.2 and 1.1 was the ability to specify how a program must be started; normal, minimized or maximized. The hotkeys definition file has been changed to accommodate for this change. Older versions of the hotkeys definition file will be automatically converted to the new format as soon as they're opened.

### What was new in version 1.1?

A couple of bugs were fixed:

- {bmc bullet.bmp} A memory leak in the TrayIcon component caused Hotkeys to consume a small amount of memory every time a hotkey was pressed. This has been fixed.
- {bmc bullet.bmp} The SendKeys didn't handle menu keys very well. For example Alt+F4 didn't close the active window. This has been fixed.

### New features in version 1.1:

- {bmc bullet.bmp} Execute multiple actions with the press of only one keystroke. This allows for scripting certain tasks (e.g. starting your e-mail program, "Sendkeying" the username and password, checking for mail and close the program again).
- {bmc bullet.bmp} ExitWindows now has the option to Restart Windows, Exit to Dos or Suspend the computer. Two new files were added to the Hotkeys package for this: HKRESTRT.EXE is a 16-bit application that restarts Windows without rebooting your machine; HKTODOS.PIF is a file that emulates the 'Restart in MS-DOS mode' option in the Windows shutdown dialog.
- {bmc bullet.bmp} A very basic and extremely small setup application with uninstall was added. This application installs Hotkeys on your machine and adds entries to the registry so that Hotkeys is started automatically every time Windows starts. It also allows for a clean uninstall.
- {bmc bullet.bmp} It is now possible to specify a delay when sending keystrokes to other applications. Especially if you are sending commands to a DOS-application, keys were sometimes faster than the program could handle, resulting in erroneous input to the program. To set a delay, simply specify {DELAY=xxxx} anywhere in the string to send.

### Other changes:

- {bmc bullet.bmp} The hotkeys definition file (Hotkeys.hkd) now has a different format to accommodate the new 'multiple commands' feature (ID for every hotkey was added). Older versions of the hotkeys definition file will be automatically converted to the new format as soon as they're opened.

### Special note for users of Hotkey version prior to 1.1

Hotkeys now comes with an installation program that copies the Hotkeys application to the directory C:\Program Files\Hotkeys. After the installation, the new version will start automatically. Unless you have previously installed Hotkeys in the aforementioned directory, Hotkeys will use the default hotkey definition file instead of the one you were using. To have Hotkeys work with your own hotkey definition file, you need to exit Hotkeys and copy your Hotkeys.hkd file to the C:\Program Files\Hotkeys directory and delete the old Hotkeys program file.

<sup>\$</sup> What's new in version 1.3?

<sup>#</sup> IDD\_WHATSNEW

<sup>k</sup> What's new?; Hotkey definition file; Setup; Sending keys to applications; Virtual clipboards

## \$<sup>#</sup>k **Select multiple commands**

This window allows you to select the commands you want to execute. You can reach this window by clicking on the button on the right of **Command IDs** textfield in the [Hotkeys editor](#). You can get more information on the 'Select commands' dialog by clicking on parts of the picture below:

```
{bmc select.shg}
```

Instead of using the 'Select commands' dialog you can also type the IDs of the actions you want to execute directly in the Command IDs textfield, separated by semi-colons. To specify a delay type 'D=xxx' where xxx is the delay in milliseconds.

\$ Select multiple commands

# IDD\_SELECT

k Execute multiple Hotkeys commands



#Lists all the available Hotkeys commands that are not yet included in 'Commands to Execute' list

#Lists the commands and delays that will be executed when the hotkey for this item is pressed

#To specify a delay between two commands, specify the delay-time (in milliseconds) in the textfield and press the Add button. To edit an existing delay, select the delay-command in the 'Commands to Execute' list, edit the value and press the Edit button. A delay is effective until another delay command is encountered

# IDD\_SEL\_DEST

# IDD\_SEL\_DELAY

# Press this button to add the selected command in the 'Available commands' list to the 'Commands to execute' list

# Press this button to add the all available commands to the 'Commands to execute' list



# Press this button to remove the selected commands from the 'Commands to execute' list

# Press this button to remove all commands from the 'Commands to execute' list

# Moves the selected commands up

# IDD\_SEL\_UP

# Moves the selected commands down

# IDD\_SEL\_DOWN

# Saves the changes made in this dialog

# IDD\_SEL\_OK

#Leaves the dialog without saving changes

# IDD\_SEL\_CANCEL